A GRU-RNN-based Language Model for Wakirike

# Abstract

In this paper, a character-based Recurrent Neural Network (RNN) language model is implemented for the Wakirike language as a strategy to address the under-resourced nature of the language for Automatic Speech Recognition (ASR) speech processing. In this work, a Gated Recurrent Unit (GRU) deep RNN Language model is prescribed as a step towards faster and resource affordable ASR systems.

The Language model implemented in this article while having a small text corpus size was found to be potentially robust enough to be comparable with the state of the art language models yielding a better relative perplexity score of 2.6 when compared to the score of 3.2 on the n-gram model with smoothing.

# Introduction

In recent times there has been a significant amount of effort in the Speech recognition research community to tackle the pragmatic reality of scarcity of speech data when building speech processing systems. Comparing to a child learning a language between the ages of 2 to 5, a child is able to grasp the fundamentals of a language by being exposed to approximately 4000 hours of speech between these ages [(Versteegh, Anguera, Jansen, & Dupoux, 2016)](https://paperpile.com/c/GaEMMA/GGfD). For automatic speech recognition systems to obtain reasonable results would require between 200 to 2000 hours of aligned speech data, that is, data having both text and speech [(Hannun et al., 2014)](https://paperpile.com/c/GaEMMA/68o9). Low resource speech recognition is therefore significant for two reasons. The first is the ability to build and reproduce speech recognition/speech processing systems for a variety of uses and languages; the second is for the purpose of optimising resources within the various domains being used by these speech systems.

Broadly speaking a language can be described as low resourced, resource-poor or low-data [(Besacier, Barnard, Karpov, & Schultz, 2014)](https://paperpile.com/c/GaEMMA/VDxv) given that one or more of the following components is either limited or absent: electronic resources, an orthographic writing system, transcribed and recorded speech data, dictionary and other linguistic resources such as a phonetic dictionary and other related linguistic descriptions. This paper focuses on Wakirike language which is especially deficient in transcribed speech data. There exists a limited language dictionary for Wakirike language from which a phonetic dictionary can be derived.

A key aspect of speech recognition systems which all modern state-of-the-art Automatic Speech Recognition (ASR) systems rely on is the language model. The language model of an ASR system is a semantic bridge for the ASR system to form meaningful utterances from words and syllables detected or recognised. Language models have many uses in natural language applications such as word prediction, spell checking, optical character recognition etc. The model described in this paper builds a character-based recurrent neural network which was found to be a capable model and yet modestly resourced. Traditionally, language models derive language-semantic information by counting word occurrences and word-pair occurrences [(Allen, 1995)](https://paperpile.com/c/GaEMMA/EKpS)[(Jelinek, 1976)](https://paperpile.com/c/GaEMMA/RxXH). The recurrent neural network on the other hand builds an internal representation of the language based on long-term conditional relationships between successive characters in the training corpus [(Mikolov, Deoras, Kombrink, Burget, & Cernock, 2011)](https://paperpile.com/c/GaEMMA/YV4D). From a semantic stance, it is more logical to create a word-based language model than a character based language model. However in a resource lacking scenario, there are many gains from using a character-based model when compared to a word-based model. This is due to the fact that the data preparation process requires processing of the raw corpus alone without having a separate language vocabulary. In addition, the character based model does not suffer from out-of-vocabulary words because the character-set is a well defined and fixed set while the word vocabulary set is potentially infinite as new words are coined within various language-modifying circumstances.

The results showed that for a small corpus of about one hundred thousand words, a comparable language model could be built. The next section surveys various language model techniques used in Automatic Speech Recognition systems and in the section afterwards is a description of the recurrent neural network (RNN) technique used in this paper as well as the corpus used to create the language model. The fourth section is a presentation of the results of the experiments performed and an evaluation and conclusion presented in the last section.

# Literature Review

Statistical speech recognition is made possible by breaking down an intractable probability distribution of words given acoustic data by applying Bayesian logic. The result of this simplification is a tractable two step procedure where the acoustic probability distribution is multiplied by the probability of words known as a language model [(Gales & Young, 2008; Mikolov et al., 2011)](https://paperpile.com/c/GaEMMA/YV4D+XIqy). The probability distribution of words or language model has been traditionally obtained by counting words and using the Markov chain simplification within a Maximum Likelihood Estimation framework. Best performance of this framework known as the n-gram model is achieved using various smoothing techniques [(Chen & Goodman, 1999)](https://paperpile.com/c/GaEMMA/Y8KI).

As computing power and availability of big data became ubiquitous, the applications of neural network to more areas of research have been seen to be an alternative approach to conventional methods of machine learning and especially pattern recognition. Because the statistical language model attempts to determine the pattern of a language through conditional probabilities between words, it is also possible to achieve the same using deep neural networks.

In 2003, Bengio et.al. [(Bengio, Schwenk, Senécal, Morin, & Gauvain, 2003)](https://paperpile.com/c/GaEMMA/eAQr) proposed a language model based on neural multi-layer perceptrons (MLPs). These MLP language models resort to a distributed representation of all the words in the vocabulary such that the probability function of the word sequences is expressed in terms of these word-level vector representations. The result of the MLP-based language models was found to be, in cases for models with large parameters, performing better than the traditional n-gram models.

Improvements over the MLPs still using neural networks over the next decade include works of [(Mikolov et al., 2011)](https://paperpile.com/c/GaEMMA/YV4D),[(Sutskever, Vinyals, & Le, 2014)](https://paperpile.com/c/GaEMMA/dm4u),[(Hannun et al., 2014; Luong, Socher, & Manning, 2013; Sutskever et al., 2014)](https://paperpile.com/c/GaEMMA/dm4u+68o9+BsjH) involved the utilisation of deep neural networks for estimating word probabilities in a language model. While a Multi-Layer Perceptron consists of a single hidden layer in addition to the input and output layers, a deep network in addition to having several hidden layers are characterised by complex structures that render the architecture beyond the basic feed forward nature where data flows from input to output hence in the RNN architecture we have some feedback neurons as well. Furthermore, the probability distributions in these deep neural networks were either based upon word or sub-word models this time having representations which also conveyed some level of syntactic or morphological weights to aid in establishing word relationships. These learned weights are referred to as token or unit embeddings.

For the neural network implementations so far seen, a large amount of data is required due to the nature of words to have large vocabularies, even for medium-scale speech recognition applications. Yoon Kim et. al. [(Kim, Jernite, Sontag, & Rush, 2016)](https://paperpile.com/c/GaEMMA/Ak5w) on the other hand took a different approach to language modelling taking advantage of the long-term sequence memory of long-short-term memory cell recurrent neural network (LSTM-RNN) to rather model a language based on characters rather than on words. This greatly reduced the number of parameters involved and therefore the complexity of implementation. This method is particularly of interest to this article and forms the basis of the implementation described in this article due to the low resource constraints imposed when using a character-level language model.

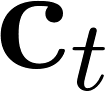
Other low resource language modelling strategies employed for the purpose of speech recognition was demonstrated by [(Xu, 2013)](https://paperpile.com/c/GaEMMA/ModO). The language model developed in that work was based on phrase-level linguistic mapping from a high resource language to a low resource language using a probabilistic model implemented using a weighted finite state transducer (WFST). This method uses WFST rather than a neural network due to scarcity of training data required to develop a neural network. However, it did not gain from the high nonlinearity ability of a neural network model to discover hidden patterns in data, being a shallower machine learning architecture.

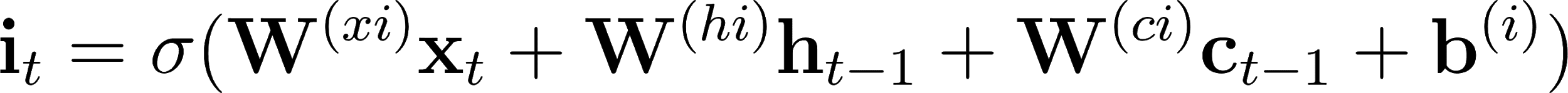
The method employed in this article uses a character-based Neural network language model that employs an LSTM network similar to that of [(Kim et al., 2016)](https://paperpile.com/c/GaEMMA/Ak5w) on the Wakirike language which is a low resource language bearing in mind that the character level network will reduce the number of parameters required for training just enough to develop a working language model for the purpose of speech recognition. The description of the data and procedure used to develop the language model is discussed in the next section.

# The GRU Cell Recurrent Neural Network Configuration

Neural networks have become increasingly popular due to their ability to model non-linear system dynamics. Since their inception, there have been many modifications made to the original design of having linear affine transformations terminated with a nonlinear functions as the means to capture both linear and non-linear features of the target system. In particular, one of such neural network modifications, namely the recurrent neural network, has been shown to overcome the limitation of varying lengths in the inputs and outputs of the classic feed-forward neural network. In addition the RNN is not only able to learn non-linear features of a system but has also been shown to be effective at capturing the patterns in sequential data.

This work draws upon the premise that the grammar of a language is expressed in the character sequence pattern ultimately revealed in the words rendered by the character sequences. Therefore, abstract grammar rules can be extracted and learned by a character-based RNN neural network. Specialised implementations of the RNN called the Long Short Term Memory (LSTM) and also the Gated Recurrent Unit are designed to capture patterns over particularly long data sequences and are thus, ideal candidates for generating character sequences while preserving syntactic language rules in the words formed from generated character sequences. These long-term relationships and patterns are learned by the neural network model from the training data it ingests.

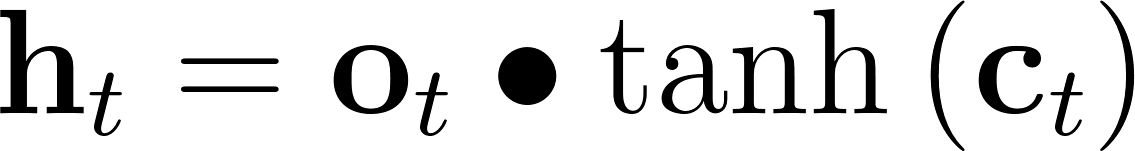
The internal structure and working of the LSTM cell is documented by its creators in [(Sak, Senior, & Beaufays, 2014)](https://paperpile.com/c/GaEMMA/ZAWC). The ability to recall information over extended sequences results from the internal gated structure which performs a series of element wise multiplications on the inputs and internal state of the LSTM cell at each time step. In addition to the output neurons which in this text we refer to as the write gate and denote as the current cell state, [](https://www.codecogs.com/eqnedit.php?latex=%5Cmathbf%7Bc%7D_t%0), three additional gates (comprising a neural network sub-layer) located within the LSTM cell are the input gate, the forget gate and the output gate. Together with the initial current state cell these gates along with the current-state cell itself enable the LSTM cell architecture to store information, forward information, delete information and receive information. Generally however, the LSTM cell looks like a regular feed-forward network having a set of neurons capped with a nonlinear function. The recurrent nature of the network arises, however due to the fact that the internal state of the RNN cell is rerouted back as an input to the RNN cell or input to the next cell in the time-series give rise to sequence memory within the LSTM architecture. Mathematically, these gates are formulated as follows:

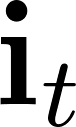
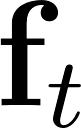
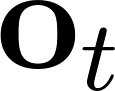
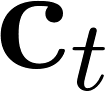
[](https://www.codecogs.com/eqnedit.php?latex=%5Cmathbf%7Bi%7D_t%3D%5Csigma(%5Cmathbf%7BW%7D%5E%7B(xi)%7D%5Cmathbf%7Bx%7D_t%2B%5Cmathbf%7BW%7D%5E%7B(hi)%7D%5Cmathbf%7Bh%7D_%7Bt-1%7D%2B%5Cmathbf%7BW%7D%5E%7B(ci)%7D%5Cmathbf%7Bc%7D_%7Bt-1%7D%2B%5Cmathbf%7Bb%7D%5E%7B(i)%7D)%0) - - - (1)

[](https://www.codecogs.com/eqnedit.php?latex=%5Cmathbf%7Bf%7D_t%3D%5Csigma(%5Cmathbf%7BW%7D%5E%7B(xf)%7D%5Cmathbf%7Bx%7D_t%2B%5Cmathbf%7BW%7D%5E%7B(hf)%7D%5Cmathbf%7Bh%7D_%7Bt-1%7D%2B%5Cmathbf%7BW%7D%5E%7B(cf)%7D%5Cmathbf%7Bc%7D_%7Bt-1%7D%2B%5Cmathbf%7Bb%7D%5E%7B(f)%7D)%0) - - - (2)

[](https://www.codecogs.com/eqnedit.php?latex=%5Cmathbf%7Bc%7D_t%3D%5Cmathbf%7Bf%7D_t%5Cbullet%5Cmathbf%7Bc%7D_%7Bt-%201%7D%2B%5Cmathbf%7Bi%7D_t%5Cbullet%5Ctanh(%5Cmathbf%7BW%7D%5E%7B(xc)%7D%5Cmathbf%7Bx%7D_t%2B%5Cmathbf%7BW%7D%5E%7B(hc)%7D%5Cmathbf%7Bh%7D_%7Bt-1%7D%2B%5Cmathbf%7Bb%7D%5E%7B(c)%7D)%0) - - - (3)

[](https://www.codecogs.com/eqnedit.php?latex=%5Cmathbf%7Bo%7D_t%3D%5Csigma(%5Cmathbf%7BW%7D%5E%7B(xo)%7D%5Cmathbf%7Bx%7D_t%2B%5Cmathbf%7BW%7D%5E%7B(ho)%7D%5Cmathbf%7Bh%7D_%7Bt-1%7D%2B%5Cmathbf%7BW%7D%5E%7B(co)%7D%5Cmathbf%7Bc%7D_%7Bt-1%7D%2B%5Cmathbf%7Bb%7D%5E%7B(o)%7D)%0) - - - (4)

[](https://www.codecogs.com/eqnedit.php?latex=%5Cmathbf%7Bh%7D_t%3D%5Cmathbf%7Bo%7D_t%5Cbullet%5Ctanh%7B(%5Cmathbf%7Bc%7D_t)%7D%20%0) - - - (5)

The gates in the above formula are illustrated in Figure ~\ref{fig1:lstmcell}. [](https://www.codecogs.com/eqnedit.php?latex=%5Cmathbf%7Bi%7D_t%0) represents the input gate, [](https://www.codecogs.com/eqnedit.php?latex=%5Cmathbf%7Bf%7D_t%0) is the forget gate and [](https://www.codecogs.com/eqnedit.php?latex=%5Cmathbf%7Bo%7D_t%0) represents the output gate. At each of these gates therefore, the inputs consisting of hidden states in addition to the regular inputs are multiplied by a set of weights and passed through a soft-max function. These weights during training learn whether the gate will, during inference, open or not. In summary, the input gate tells the LSTM not whether or not to receive new information, the forget gate determines whether the current information it already has from the previous step should be kept or dropped and the output gate determines what should be forwarded to the next LSTM cell. Note also that the LSTM has two sigmoid (tanh) activation functions utilised at the input and output of the current cell [](https://www.codecogs.com/eqnedit.php?latex=%5Cmathbf%7Bc%7D_t%0).

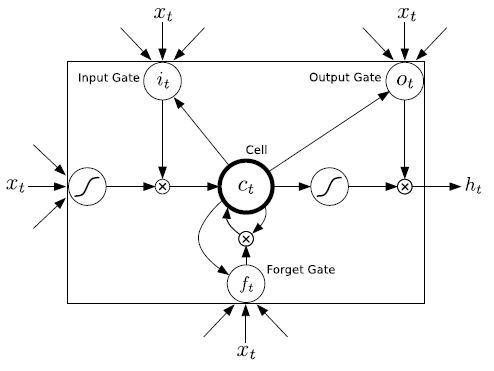


Figure 1: LSTM Cell \ref{}

# Experiments

## Dataset preparation

The Wakirike New Testament Bible served as the source of data for the deep neural network training. There is no readily available soft or on-line copy of the Wakirike new testament bible. As such, the Wakirike New Testament Bible was typed to form a text corpus, giving rise to a complete corpus word size of 668,522 words and a character count of 6,539,176 characters. The dataset was then divided into 11 parts. Two parts dedicated for testing and validation and the remaining nine parts were used for training.

The unicode representations of the character set consisting of letters and punctuation marks are one-hot encoded and batched for sequential input, each batch having a character sequence length of 30 characters.

## [GRU](https://arxiv.org/pdf/1406.1078v3.pdf) Training

The modified LSTM RNN known as the Gated Recurrent Unit (GRU) is employed for the neural network model built in this work, in order to optimise network performance while conserving computation resources. GRUs have been shown to give similar performance to regular LSTMs however, with a lighter system resource footprint [(Cho et al., 2014)](https://paperpile.com/c/GaEMMA/MExf). The GRU RNN used to train the Wakirike text corpus comprised an internal network size of 512 nodes for each layer and was 3 layers deep. Externally, 30 GRUs represented the number of recurrent connections each connection representing a time step bearing contextual for the recurrent input sequence.

To mitigate for overfitting, due to the multi-layered high-dimensional depth of this neural network, a small learning rate of 0.001 was used. To further marginalise overfitting the popular and effective dropout (Srivastava et al., 2014) method for regularising deep neural networks kept at 20% such that only 80% of neural network activations are propagated from one layer to the next, whereas the remaining 20% were randomly zeroed out.

## Output Language Generation

The neural network was trained for 10 epochs and achieved a prediction accuracy of 85% on held-out data. With this GRU character-based language model, it is possible to seed this network with an input character and select from the top-N candidates thus causing the Neural network to generate its own sentences. In this scenario, the network is said to perform language generation by immanently constructing its own sentences. The generated language output from the GRU language model was found to be intelligible and a reflection of the overall context of the training data.

A clever use of this new corpus generated by the GRU language model of this work was to determine a word-based perplexity metric for the GRU neural language model. In this work, the word-based perplexity metric was achieved from the output language generated by first estimating the word based perplexity on the training data. The same perplexity calculation was then used on the generated neural language model corpus. The corpus size of the neural language model was made to be equivalent to that of the training data, that is containing 6,539,176 characters. The perplexity calculation was based on a modified Kneser-Key 5-gram model with smoothing \citep{Heafield-estimate}. The results discussed below showed that the LSTM model generated a superior model compared to the n-gram model that better matched the training data.

# 

# The evaluation of the GRU language model of the Wakirike language was performed using a perplexity measurement metric. The Perplexity metric applies the language model to a test dataset and measures how probable the test dataset is. Perplexity is a relative measure given by the formula:

# - - - (6)

# - - - (7)

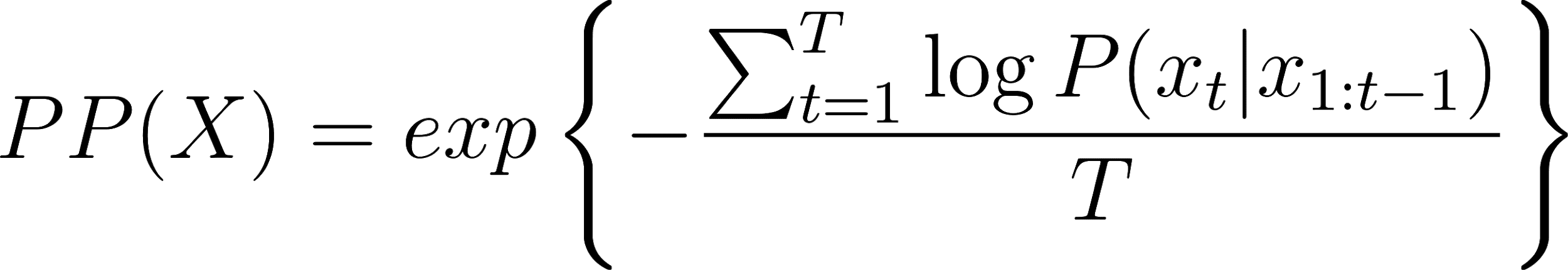
# Where are the sequence of words. The language model with the lower relative perplexity score is therefore expected to yield better approximation of the data when applied to unseen data generally.

# 

# Discussion

The result of the training of the GRU-Cell Recurrent Neural Network on low-resourced Wakirike Language gave impressive and intelligible results and showed better results when measured with standard n-gram language models. The results showed that it is indeed possible to derive a language model using a GRU-cell RNN on a low resource character sequence corpus for the Wakirike language.

A character based perplexity metric is possible using the negative log likelihood of the character sequence.

[](https://www.codecogs.com/eqnedit.php?latex=PP(X)%3Dexp%5Cleft%5C%7B-%5Cfrac%7B%5Csum_%7Bt%3D1%7D%5ET%5Clog%20P(x_t%7Cx_%7B1%3At-1%7D)%7D%7BT%7D%5Cright%5C%7D%0) - - - (8)

However, our base-line language model is a 5-gram word-based language model. Therefore, comparing a word based model to a character based model requires a conversion step. In this work, the conversion step involved using the GRU language model generated a corpus which was rescored by re-estimating with a 5-gram word-based language model described in the previous section.

Table 1 shows the Results of the Perplexity model of the GRU Wakirike Language model and an equivalent 5-gram Language model with interpolation and Keysner smoothing [(Chen & Goodman, 1999)](https://paperpile.com/c/GaEMMA/Y8KI) for various lengths of the held-out data.

### Table 1

|  |  |  |
| --- | --- | --- |
| Language model | Perplexity | |
| Held-out data size(characters) | 998 | 99 |
| GRU RNN | 1.6398 | 1.7622 |
| 5-gram with Keysner Smoothing and interpolation | 1.8046 | 1.9461 |

It can be inferred that the GRU character-model developed has an improved language model and because it is based on a character-model, which is fine-grained when compared to a word model, it is likely to generalise data better when used in practice is and less biased than a word-based model. This can be observed from the fact that the output corpus produced a larger vocabulary size.

# Conclusion

There is a strong need to consolidate and optimise the research being carried out in automatic speech recognition. An exciting aspect of the research is the need to verify methods on different languages not previously exposed to the current ASR methods. This paper presented an LSTM language model for Wakirike language as a step towards low resource speech recognition. The LSTM language model, was optimised using a bespoke character-based LSTM model to compensate for the low resource text data seen in Wakirike as a language. At the same time, the LSTM RNN model produced a fine-grained language model as opposed to traditional word-based models or other statistical models and was able to produce acceptable results. Having seen the perplexity measurement that performed better than the statistical language model counterparts, the language model produced in this work can serve as a basis for a more elaborate generative model which can in turn serve as a substrate for generative adversarial techniques. Conversely, generative adversarial techniques [(Chen & Goodman, 1999; Goodfellow et al., 2014)](https://paperpile.com/c/GaEMMA/Y8KI+1ukC), which is the subject of a future work, presents a valuable method to tackle the low resource ASR challenge.

# Acknowledgements

The authors would like to thank Andrej Karpathy and Martin Gomer for their character LSTM implementations from which this work was adapted.

# References

[Allen, J. (1995). *Natural Language Understanding*. Addison-Wesley.](http://paperpile.com/b/GaEMMA/EKpS)

[Bengio, Y., Schwenk, H., Senécal, J.-S., Morin, F., & Gauvain, J.-L. (2003). Neural Probabilistic Language Models. *Innovations in Machine Learning*, pp. 137–186. https://doi.org/](http://paperpile.com/b/GaEMMA/eAQr)[10.1007/3-540-33486-6\_6](http://dx.doi.org/10.1007/3-540-33486-6_6)

[Besacier, L., Barnard, E., Karpov, A., & Schultz, T. (2014). Automatic speech recognition for under-resourced languages: A survey. *Speech Communication*, Vol. 56, pp. 85–100. https://doi.org/](http://paperpile.com/b/GaEMMA/VDxv)[10.1016/j.specom.2013.07.008](http://dx.doi.org/10.1016/j.specom.2013.07.008)

[Chen, S. F., & Goodman, J. (1999). An empirical study of smoothing techniques for language modeling. *Computer Speech & Language*, Vol. 13, pp. 359–393. https://doi.org/](http://paperpile.com/b/GaEMMA/Y8KI)[10.1006/csla.1999.0128](http://dx.doi.org/10.1006/csla.1999.0128)

[Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. https://doi.org/](http://paperpile.com/b/GaEMMA/MExf)[10.3115/v1/d14-1179](http://dx.doi.org/10.3115/v1/d14-1179)

[Gales, M., & Young, S. (2008). *The Application of Hidden Markov Models in Speech Recognition*. Now Publishers Inc.](http://paperpile.com/b/GaEMMA/XIqy)

[Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., … Bengio, Y. (2014). Generative adversarial nets. *Advances in Neural Information Processing Systems*, 2672–2680.](http://paperpile.com/b/GaEMMA/1ukC)

[Hannun, A., Case, C., Casper, J., Catanzaro, B., Diamos, G., Elsen, E., … Coates, A. (2014). Deep speech: Scaling up end-to-end speech recognition. *arXiv Preprint arXiv:1412.5567*.](http://paperpile.com/b/GaEMMA/68o9)

[Jelinek, F. (1976). Continuous speech recognition by statistical methods. *Proceedings of the IEEE*, Vol. 64, pp. 532–556. https://doi.org/](http://paperpile.com/b/GaEMMA/RxXH)[10.1109/proc.1976.10159](http://dx.doi.org/10.1109/proc.1976.10159)

[Kim, Y., Jernite, Y., Sontag, D., & Rush, A. M. (2016). Character-Aware Neural Language Models. *AAAI*, 2741–2749.](http://paperpile.com/b/GaEMMA/Ak5w)

[Luong, T., Socher, R., & Manning, C. D. (2013). Better word representations with recursive neural networks for morphology. *CoNLL*, 104–113.](http://paperpile.com/b/GaEMMA/BsjH)

[Mikolov, T., Deoras, A., Kombrink, S., Burget, L., & Cernock, J. (2011). Empirical evaluation and combination of advanced language modeling techniques. *Twelfth Annual Conference of the International Speech Communication Association*. Presented at the Twelfth Annual Conference of the International Speech Communication Association.](http://paperpile.com/b/GaEMMA/YV4D)

[Sak, H., Senior, A., & Beaufays, F. (2014). Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition. *arXiv Preprint arXiv:1402.1128*.](http://paperpile.com/b/GaEMMA/ZAWC)

[Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. *Advances in Neural Information Processing Systems*, 3104–3112.](http://paperpile.com/b/GaEMMA/dm4u)

[Versteegh, M., Anguera, X., Jansen, A., & Dupoux, E. (2016). The Zero Resource Speech Challenge 2015: Proposed Approaches and Results. *Procedia Computer Science*, Vol. 81, pp. 67–72. https://doi.org/](http://paperpile.com/b/GaEMMA/GGfD)[10.1016/j.procs.2016.04.031](http://dx.doi.org/10.1016/j.procs.2016.04.031)

[Xu, P. (2013). *Cross-lingual language modeling for low-resource speech recognition*. https://doi.org/](http://paperpile.com/b/GaEMMA/ModO)[10.14711/thesis-b1180211](http://dx.doi.org/10.14711/thesis-b1180211)